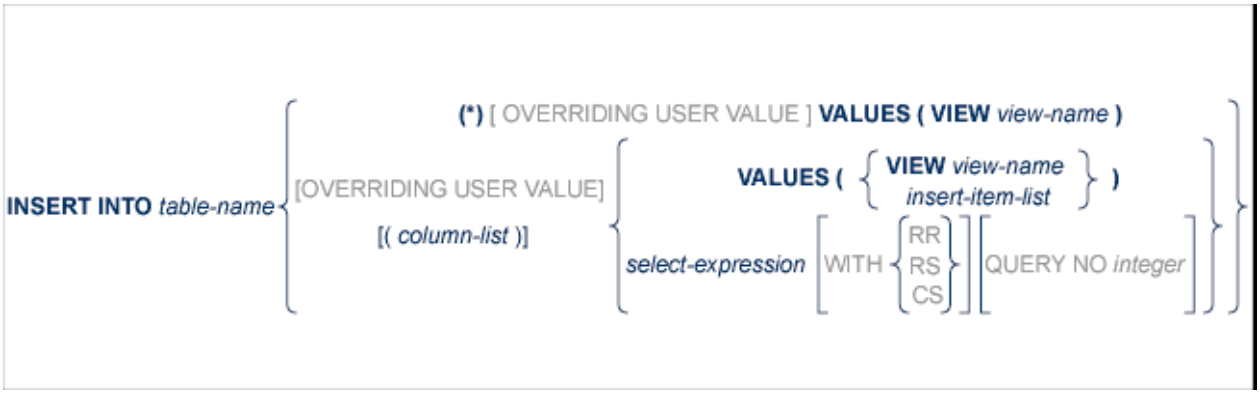
INSERT



Function

The SQL INSERT statement is used to add one or more new rows to a table.

INTO Clause

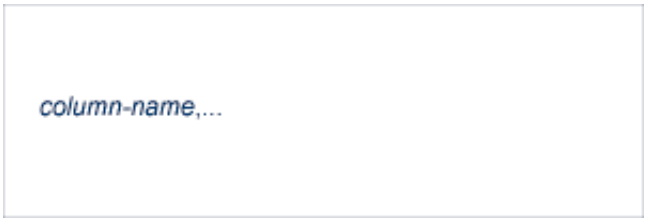
In the INTO clause, the table is specified into which the new rows are to be inserted.

See further information on *table-name*.

OVERRIDING USER VALUE

OVERRIDING USER VALUE belongs to the SQL Extended Set. This clause causes the value specified in the VALUES clause or produced by a fullselect for a column that is defined as GENERATED ALWAYS to be ignored.

column-list



In the *column-list*, one or more columns can be specified, which are to be supplied with values in the row currently inserted.

If a *column-list* is specified, the sequence of the columns must match with the sequence of the values either specified in the *insert-item-list* or contained in the specified view (see below).

If the *column-list* is omitted, the values in the *insert-item-list* or in the specified view are inserted according to an implicit list of all the columns in the order they exist in the table.

VALUES Clause

With the VALUES clause, you insert a *single* row into the table. Depending on whether an asterisk (*) or a *column-list* has been specified, the VALUES clause can take one of the following forms:

VALUES Clause with Preceding Asterisk Notation

If asterisk notation is specified, a view **must** be specified in the VALUES clause. With the field values of this view, a new row is inserted into the specified table using the field names of the view as column names of the row.

VALUES Clause with Preceding Column List

If a *column-list* is specified and a view is referenced in the VALUES clause, the number of items specified in the column list must correspond to the number of fields defined in the view.

If no *column-list* is specified, the fields defined in the view are inserted according to an implicit list of all the columns in the order they exist in the specified table.

insert-item-list

In the *insert-item-list*, you can specify one or more values to be assigned to the columns specified in the *column-list*. The sequence of the specified values must match the sequence of the columns.

If no *column-list* is specified, the values in the *insert-item-list* are inserted according to an implicit list of all the columns in the order they exist in the table.

The values to be specified in the *insert-item-list* can be *constants*, *parameters*, *special-registers* or NULL.

See the section Basic Syntactical Items for information on *view-name*, *constant* and *parameter*. See also the information on *special-register*.

If the value NULL has been assigned, this means that the addressed field is to receive no value (not even the value "0" or "blank").

Example - INSERT Single Row:

```
...
INSERT INTO SQL-PERSONNEL (NAME,AGE)
  VALUES ('ADKINSON',35)
 ...
```


select-expression

With a *select-expression*, you insert *multiple* rows into a table. The *select-expression* is evaluated and each row of the result table is treated as if the values in this row were specified as values in a VALUES clause of a single-row INSERT operation.

See further information on *select-expression*.

Example - INSERT Multiple Rows:

```
...
INSERT INTO SQL-RETIREE (NAME,AGE,SEX)
  SELECT LASTNAME, AGE, SEX
  FROM SQL-EMPLOYEES
  WHERE AGE > 60
...
```

Note:
The number of rows that have actually been inserted can be ascertained by using the system variable *ROWCOUNT (see Natural System Variables documentation).

WITH isolation level clause

The WITH isolation level clause belongs to the SQL Extended Set and allows the explicit specification of the isolation level used when locating the rows to be inserted.

QUERY NO Clause

The QUERYNO clause belongs to the SQL Extended Set and explicitly specifies the number to be used in EXPLAIN output and trace records for this statement.

